```
module MusicalChairs(clk, en, qtr);
  input clk;
  input en;
  output qtr;
  reg [1:0] Q;
  reg [1:0] D;

// Initial State
  initial Q = 2'b00;

// Output logic
  assign qtr = Q[1] && Q[0];

// Version 1: Describe next-state logic
//------------------------------------
// FF D inputs (Combinational)
  assign D[1] = en && (Q[1] ^ Q[0]) || ~en && Q[1];
  assign D[0] = en ^ Q[0];

// Next-State Equations
  always @(posedge clk)
  begin
    Q[1] <= D[1];
    Q[0] <= D[0];
  end
```

2 bits vector

output

we define the input and output of flip-flop as register (state holding element)

corresponding input

initialize the vector we defined for the flip flops

what we say here is we will do it repeatedly.

the equation we got from the lecture

assignment to the next state

"<=" is sequential assignment.

these two assignment happens concurrently

means repeat forever

Q: How the state change

A: change on a positive edge of the clock

```
module Testbench();
  reg clk; define clock as register
  reg en;
  wire qtr;

  MusicalChairs MC(clk,en, qtr);

// clk   100ns clock period
  always #50 clk = ~clk;

// in
  initial
  begin
    clk = 1;
    en = 0;
    #75;
    en = 1;
    #500
    en = 0;
    #125;
```

```
// Versioon 2: Describe state transitons
//-------------------------------------


// State Update (DFF Characteristic Equation)
  always @(posedge clk)
    Q <= D;

// Assign state labels
  parameter ZERO  = 2'b00;
  parameter ONE   = 2'b01;
  parameter TWO   = 2'b10;
  parameter THREE = 2'b11;

// State Transitions

  always @(en, Q)
  begin
    case(Q)

      ZERO:   if (en)
                D = ONE;
              else
                D = ZERO;
      ONE:    if (en)
                D = TWO;
              else
                D = ONE;
      TWO:    if (en)
                D = THREE;
              else
                D = TWO;

      THREE:  if (en)
                D = ZERO;
              else
                D = THREE;
    endcase
  end
//


endmodule
```

UM EECS 270 F22
Introduction to Logic Design

11. Design of Sequential Circuits

Sequential Circuit Design

- Goal: Design a sequential circuit that satisfies the requirements of the given problem description (spec.)
- Follow sequential circuit analysis steps in reverse!
  (more or less)

1) (optional) Construct state diagram — *"Art of design"*
2) Construct state/output table
3) Create state assignments
4) Create transition/output table — *"Turn the crank"*
5) Choose FF type
6) Construct **excitation/output table**
   - Similar to transition/output table
7) Find excitation and output logic equations

Sequential Design Example

- Problem description: design a Moore sequential circuit with one input IN and one output OUT, such that OUT is one iff IN is 1 for three consecutive clock cycles
- State table:

	IN		
S	0	1	OUT
zero1s	zero1s	one1	0
one1	zero1s	two1s	0
two1s	zero1s	three1s	0
three1s	zero1s	three1s	1
	S^+		

- State Assignments
  - What is the minimum number of state variables needed to encode four states?

$$2$$

  - In general, if we have *n* states, what is the minimum number of state variables needed to encode those states?

$$\lceil \log_2 n \rceil$$

State name	Q_1	Q_0
zero1s	0	0
one1	0	1
two1s	1	0
three1s	1	1

These state assignments may seem rather arbitrary – that's because they are! We will soon see the impact that state assignments have on our final circuit…

- Transition/output table

State/Output Table:

S	IN 0	IN 1	OUT
zero1s	zero1s	one1	0
one1	zero1s	two1s	0
two1s	zero1s	three1s	0
three1s	zero1s	three1s	1
	S^+		

symbolic

State Assignments

State name	Q_1	Q_0
zero1s	0	0
one1	0	1
two1s	1	0
three1s	1	1

Transition/Output Table:

Q_1	Q_0	IN 0	IN 1	OUT
0	0	00	01	0
0	1	00	10	0
1	0	00	11	0
1	1	00	11	1
		Q_1^+	Q_0^+	

- Choose FF type:
  - Using D flip-flops will simplify things (as we'll see below…)

easiest, because $Q^+ = D$

- Excitation table
  - Shows *FF input values required to create next state values* for every current state/input combination
  - If we're designing with D FFs, entries in excitation/output table are the same as those in transition/output table!
    - Because of D FF characteristic equation: Q+ = D

Q_1	Q_0	IN 0	IN 1	OUT
0	0	00	01	0
0	1	00	10	0
1	0	00	11	0
1	1	00	11	1
		D_1	D_0	

$D_1 = Q_1^+$ = function of (Q_1, Q_0, IN)

$D_0 = Q_0^+$ = function of (Q_1, Q_0, IN)

- Excitation Logic

By looking at D_1: ① we find when "$IN=0$", $D_1=0$
② 当 "$IN=1$" 时 only when $Q_1=Q_0=0$ 时, $D_1=0$

Excitation/Output Table:

Q_1	Q_0	IN = 0	IN = 1	OUT
0	0	0	1	0
0	1	0	1	0
1	0	0	1	0
1	1	0	1	1
		D_1	D_0	

$$D_1 = IN' \cdot 0 + IN \cdot (Q_1' \cdot Q_0')' = IN \cdot (Q_1 + Q_0)$$

$$D_0 = IN' \cdot 0 + IN \cdot (Q_1' \cdot Q_0)' = IN \cdot (Q_1 + Q_0')$$

- Output Logic

$$OUT = Q_1 \cdot Q_0$$

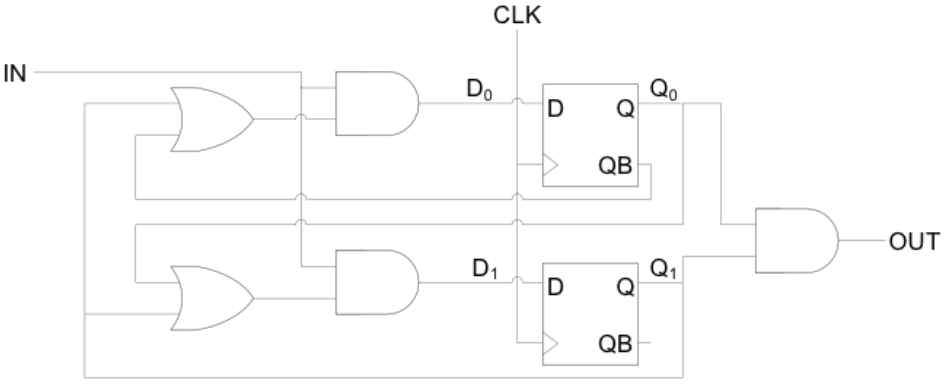
- Circuit:

Excitation Equations:

$$D_1 = IN \cdot (Q_1 + Q_0)$$
$$D_0 = IN \cdot (Q_1 + Q_0')$$

Output Equation:

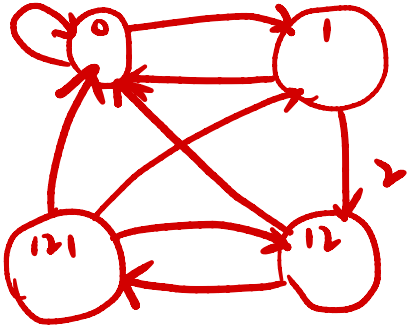
$$OUT = Q_1 \cdot Q_0$$



In Class Exercise

- Design a state/output table for the following problem specification:

Combination lock: Two inputs, X and Y, encode a binary number between 0 and 3 (X is MSB, i.e., XY = 10 →2). A single output signal UNLOCK should be set to 1 iff the sequence 1, 2, 1 occurs on the inputs in three consecutive clock cycles



Schlage BE365CAM619 Be365 Cam

https://www.amazon.com/gp/product/B000NJJ1MQ/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1

Apps | Google+ | DJIA | MLibrary | Canvas | UGRAB | Admissions | teaching-eecs-270-... | Travel and Expense ... | Other bookmarks

Tools & Home Improvement › Hardware › Door Hardware & Locks › Deadbolts

You purchased this item on March 3, 2019.
Style: **Camelot Keypad** | Color Name: **Satin Nickel** | Product Packaging: **Standard Packaging** | View this order
Try, "Alexa, reorder schlage keyless lock"



Roll over image to zoom in

Schlage BE365CAM619 Be365 Camelot Keypad Deadbolt, Satin Nickel

by Schlage Lock Company
4.4 stars | 5,203 customer reviews | 1000+ answered questions
Amazon's Choice for "schlage keyless lock"

Price: $91.99 prime

Style: **Camelot Keypad**

Camelot Keypad | Camelot Keypad + Handleset | Plymouth Keypad

Color Name: **Satin Nickel**

Product Packaging: **Standard Packaging**

Standard Packaging | Ecommerce Packaging

Setup options: **Get expert setup** Details

Without expert setup | Expert setup +$108.17 per unit

What's included

- Create and delete access codes for trusted friends and family (up to 19)
- Installs in minutes - no wiring needed;Door thickness range: 1-5/8 inches to 2 inches thick (41mm-51mm) standard,Thick door kit available
- Guaranteed to fit on standard doors. Backset of universal latches and deadbolts fits 2 to 0.375 inch or 2 to0.75 inch backsets
- Silicone-coated keypad prevents numbers from wearing off
- Pairs with our most popular trim styles to coordinate the look throughout your home

› See more product details

Buy New
$91.99
prime

FREE Delivery by **Tuesday** if you order within 7 hrs 49 mins. Details

In Stock.

Qty: 1

Add to Cart

Buy Now

Ships from and sold by Amazon.com. Gift-wrap available.

Item arrives in packaging that reveals what's inside. To hide it, choose **Ship in Amazon packaging** at checkout.

Add a Protection Plan:
4-Year Protection for $12.99
3-Year Protection for $7.99

Deliver to Karem - Ann Arbor 48103

Buy Used

Type here to search

ENG 4:10 PM 3/17/2019

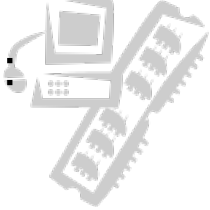
In Class Exercise

- Design a state/output table for the following problem specification:

Combination lock: Two inputs, X and Y, encode a binary number between 0 and 3 (X is MSB, i.e., XY = 10 →2). A single output signal UNLOCK should be set to 1 iff the sequence 1, 2, 1 occurs on the inputs in three consecutive clock cycles

		X Y				
	S	00	01	10	11	UNLOCK
got nothing	A	A	B	A	A	0
got "1"	B	A	B	C	A	0
got "1,2"	C	A	D	A	A	0
got "1,2,1"	D	A	B	C	A	1
		S^+				

FSM Transition List Design: 50s Vending Machine

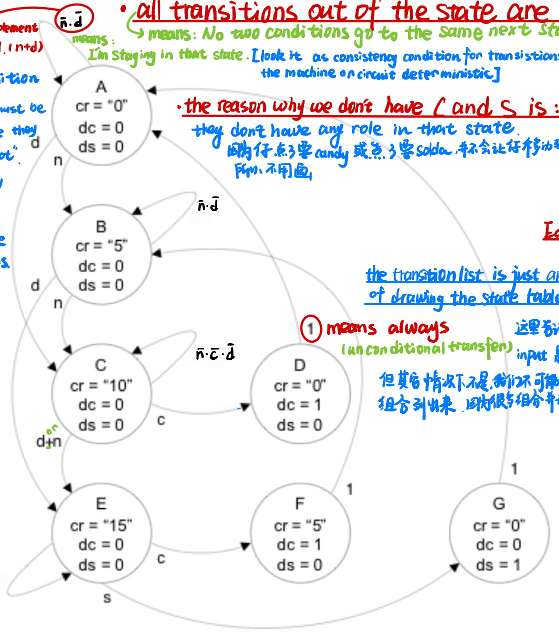


- Inputs
  - **d**: asserted when user inserts dime
  - **n**: asserted when user inserts nickel
  - **c**: asserted when user presses candy button
  - **s**: asserted when user presses soda button
- Outputs
  - **dc**: dispenses candy when asserted
  - **ds**: dispenses soda when asserted
  - **cr**: 4-bit unsigned number, represents the user's credit
- Specifications
  - All inputs are one-hot
  - Candy costs 10 cents, soda costs 15 cents
  - Money need only be counted up to 15 cents

dn s c has 16 combinations, but you can not press two options at the same time.

Vending Machine State Diagram and Transition List

- Not all of the input are indicated on the arrows
- all transitions out of the state are mutually exclusive
  → means: No two conditions go to the same next state



[look it as consisteng condition for transistions to make the machine on circuit deterministic]

- the reason why we don't have C and S is:
  they don't have any role in that state.
  因为仔点了要candy或点了要solda,并不会让仔移动到下一个state 所以,不用画

Output Table

	$Q_2Q_1Q_0$	cr	dc	ds
A	0 0 0	"0"	0	0
B	0 0 1	"5"	0	0
C	0 1 0	"10"	0	0
D	0 1 1	"0"	1	0
E	1 0 0	"15"	0	0
F	1 0 1	"5"	1	0
G	1 1 0	"0"	0	1

Each row in the List is an arrow in the diagrame

the transition list is just an alternative way of drawing the state table.

(1) means always (unconditional transfer)

这里虽许仔 input 是one-hot

但其它情况下不是,我们不可能把所有组合列出来,因为很多组合并无意义.

Transition List

2^{nd} collumn (n input) If the input is n. "one-hot"

S	$Q_2Q_1Q_0$	Transition Expression	S^+	$Q_2^+Q_1^+Q_0^+$
A	0 0 0	n	B	0 0 1
A	0 0 0	d	C	0 1 0
A	0 0 0	$\bar{n}\cdot\bar{d}$	A	0 0 0
B	0 0 1	n	C	0 1 0
B	0 0 1	d	E	1 0 0
B	0 0 1	$\bar{n}\cdot\bar{d}$	B	0 0 1
C	0 1 0	n+d	E	1 0 0
C	0 1 0	c	D	0 1 1
C	0 1 0	$\bar{n}\cdot\bar{d}\cdot\bar{c}$	C	0 1 0
D	0 1 1	1	A	0 0 0
E	1 0 0	c	F	1 0 1
E	1 0 0	s	G	1 1 0
E	1 0 0	$\bar{s}\cdot\bar{c}$	E	1 0 0
F	1 0 1	1	B	0 0 1
G	1 1 0	1	A	0 0 0

并不重要. 对于state的变化 很多input其是 symbolic. 因为

这里我们表示input没有用binary的方式.而是使用了

Transition List

S	$Q_2Q_1Q_0$	Transition Expression	S^+	$Q_2^+Q_1^+Q_0^+$
A	0 0 0	n	B	0 0 1
A	0 0 0	d	C	0 1 0
A	0 0 0	$\bar{n}\cdot\bar{d}$	A	0 0 0
B	0 0 1	n	C	0 1 0
B	0 0 1	d	E	1 0 0
B	0 0 1	$\bar{n}\cdot\bar{d}$	B	0 0 1
C	0 1 0	n+d	E	1 0 0
C	0 1 0	c	D	0 1 1
C	0 1 0	$\bar{n}\cdot\bar{d}\cdot\bar{c}$	C	0 1 0
D	0 1 1	1	A	0 0 0
E	1 0 0	c	F	1 0 1
E	1 0 0	s	G	1 1 0
E	1 0 0	$\bar{s}\cdot\bar{c}$	E	1 0 0
F	1 0 1	1	B	0 0 1
G	1 1 0	1	A	0 0 0

input + State variable

搞清楚 D_2, D_1, D_0 与谁有关.

$$D_2 = Q_2^+ = \bar{Q}_2\bar{Q}_1Q_0d + \bar{Q}_2Q_1\bar{Q}_0(n+d) + Q_2\bar{Q}_1\bar{Q}_0c + Q_2\bar{Q}_1\bar{Q}_0s + Q_2\bar{Q}_1\bar{Q}_0\bar{s}\bar{c}$$

$$D_1 = Q_1^+ = \bar{Q}_2\bar{Q}_1\bar{Q}_0d + \bar{Q}_2\bar{Q}_1Q_0n + \bar{Q}_2Q_1\bar{Q}_0c + \bar{Q}_2Q_1\bar{Q}_0\bar{n}\bar{d}\bar{c} + Q_2\bar{Q}_1\bar{Q}_0s$$

$$D_0 = Q_0^+ = \bar{Q}_2\bar{Q}_1\bar{Q}_0n + \bar{Q}_2\bar{Q}_1Q_0\bar{n}\bar{d} + \bar{Q}_2Q_1\bar{Q}_0c + Q_2\bar{Q}_1\bar{Q}_0c + Q_2\bar{Q}_1Q_0$$

excitation equation

Output Table

we need four bits to represent these numbers

	$Q_2Q_1Q_0$	cr	dc	ds
A	0 0 0	"0"	0	0
B	0 0 1	"5"	0	0
C	0 1 0	"10"	0	0
D	0 1 1	"0"	1	0
E	1 0 0	"15"	0	0
F	1 0 1	"5"	1	0
G	1 1 0	"0"	0	1

$$dc = Q_2Q_1Q_0 + Q_2\bar{Q}_1Q_0$$

$$ds = Q_2Q_1\bar{Q}_0$$

output equation

…

Transition List

S	$Q_2Q_1Q_0$	Transition Expression	S^+	$Q_2^+Q_1^+Q_0^+$
A	0 0 0	n	B	0 0 1
A	0 0 0	d	C	0 1 0
A	0 0 0	$\bar{n}\cdot d$	A	0 0 0
B	0 0 1	n	C	0 1 0
B	0 0 1	$n\cdot d$	B	0 0 1
C	0 1 0	c	D	0 1 1
C	0 1 0	$\bar{n}\cdot d\cdot\bar{c}$	C	0 1 0
D	0 1 1	1	A	0 0 0
E	1 0 0	c	F	1 0 1
E	1 0 0	s	G	1 1 0
E	1 0 0	$\bar{s}\cdot\bar{c}$	E	1 0 0
F	1 0 1	1	B	0 0 1
G	1 1 0	1	A	0 0 0

$$D_2 = Q_2^+ = \overline{Q}_2\overline{Q}_1Q_0d + \overline{Q}_2Q_1\overline{Q}_0(n+d) + Q_2\overline{Q}_1\overline{Q}_0c + Q_2\overline{Q}_1\overline{Q}_0s + Q_2\overline{Q}_1\overline{Q}_0\overline{s}\overline{c}$$

$$D_1 = Q_1^+ = \overline{Q}_2\overline{Q}_1\overline{Q}_0d + \overline{Q}_2\overline{Q}_1Q_0n + \overline{Q}_2Q_1\overline{Q}_0c + \overline{Q}_2Q_1\overline{Q}_0\overline{n}\overline{d}\overline{c} + Q_2\overline{Q}_1\overline{Q}_0s$$

$$D_0 = Q_0^+ = \overline{Q}_2\overline{Q}_1\overline{Q}_0n + \overline{Q}_2\overline{Q}_1Q_0\overline{n}\overline{d} + \overline{Q}_2Q_1\overline{Q}_0c + Q_2\overline{Q}_1\overline{Q}_0c + Q_2\overline{Q}_1Q_0$$

Output Table

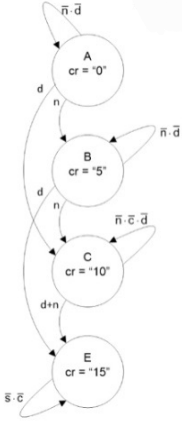
	$Q_2Q_1Q_0$	cr	dc	ds
A	0 0 0	"0"	0	0
B	0 0 1	"5"	0	0
C	0 1 0	"10"	0	0
D	0 1 1	"0"	1	0
E	1 0 0	"15"	0	0
F	1 0 1	"5"	1	0
G	1 1 0	"0"	0	1

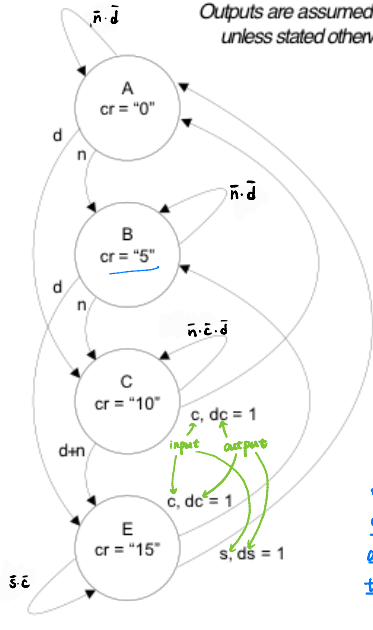
$$dc = \overline{Q}_2Q_1Q_0 + Q_2\overline{Q}_1Q_0$$

$$ds = Q_2Q_1\overline{Q}_0$$

…

50s Vending Machine: Mealy Implementation





Outputs are assumed to be 0 unless stated otherwise

Mealy outputs depend on the states and the input

The Mealy implementation uses fewer states, and therefore fewer FFs!

Q: what is the difference between state table and transition table.

A: State table is in terms of symbolic names of states (but not binary) ⟶ once you make a state assignment, the state table becomes a transition table.